DISTRIBUTED PROGRAMMING AWARENESS AMONG COMPUTER SCIENCE STUDENTS

Ogunrinde, M. A.
Department of Mathematical and Computer Sciences,
Fountain University, Osogbo, Osun State, Nigeria.
bogunrinde@gmail.com

ABSTRACT

Software development trend has tried to establish Global Software Engineering (GSE) as well as Agile methodologies such as eXtreme programming. Due to their emphasis on face-to-face communication, Agile methods seemingly present a practical dilemma for distributed development for which communication is one of the greatest challenges. However, with their equal emphasis on frequent communication, transparency of progress and short iterations, they in fact provide an effective approach to address challenges in distributed collaboration, especially when augmented with GSE best practices. Introduction, implementing and adoption of distributed agile software development in classroom environments require additional technological support for the frequent synchronous interaction across sites and is far from trivial. In this report, a well-structured online questionnaire was designed to carry out survey among the volunteered students with about 200 participants, and the results were later analyzed. The study concluded that the level of awareness of the Global Software Engineering practices is very low among students of Computer Science as well as the use of tools that facilitate it.

Keywords: Global Software Engineering, Software Development, Agile Methodology, Distributed Programming, Collaboration

1.0 INTRODUCTION

Software development is one of the strong and important aspects of Software Engineering (SE). Traditionally, software development involves a programmer developing software using a particular tool, at the same time following the development process as laid down by his/her company. Most times traditional waterfall model is being followed by the companies where contracts are signed off with the requirement document to kick start the project. The rigidness of the traditional waterfall approach to software development lead to agile software development [1]. Agile software development solves the problem of rigidity in waterfall approach by creating a User centered process of development through its methodologies such as Extreme Programming. One of the principles of Extreme Programming is pair programming which involves two programmers sitting at the same computer software artificial. Many researchers have proved that this works effectively even among computer science professionals and students [2, 3, 4, 5, 6].

Global Software Engineering GSE) is now an established trend in software development, and so are agile methods, such as Scrum [7]and Extreme Programming. Due to their emphasis on face-to-face communication, agile methods seemingly present a practical dilemma for distributed development for which communication is one of the greatest challenges. However, with their equal emphasis on frequent communication, transparency of progress and short iterations, they in fact provide an effective approach to address challenges in distributed collaboration, especially when augmented with GSE best practices [8]. Adopting agile methods in distributed settings seems to be a rising industry trend, as agile methods both provide several benefits, as well as strategies for addressing GSE challenges [9, 10]. Therefore, software engineering curricula must respond by providing students with knowledge and experiences of globally distributed agile methods and related projects. Furthermore, as software engineering researchers and educators, our community should increase its efforts in evaluating methodologies for teaching GSE e.g. [11]. Though, introduction, implementation and adoption of distributed agile software development in classroom environments may require additional technological support for the frequent synchronous interaction across sites and is far from trivial. In this paper, the report about the awareness of distributed agile software development among students is made.

1.1 Distributed Software Development

Distributed Software Development (DSD) has recently evolved, resulting in an increase in the available literature. Organizations now have a tendency to make greater development efforts in more attractive zones. The main advantage of this lies in a greater availability of human resources in decentralized zones at less cost [12, 13] Distributed development is a software development model in which IT teams spread across geographical lines collaborate on applications or various software [14], said that teams are often separated by mini-projects that are brought together for a final software build out. Teams work remotely, collaboratively and in a distributed development fashion for a number of reasons, as follows:

(i) Although team members may share similar project ideas, they may reside or work in separate locations, making in-house collaboration impossible.
(ii) Startups may use this approach to save upfront or capital costs, like facilities and hardware for team members.
(iii) Team members may want or need to work from home, or relocation may not be an option.

(iv) Globalization and hiring IT staff in third-world countries cuts overhead costs [14].

2.0 RELATED WORKS

In a work titled Student's Awareness of Cloud Computing: Case Study Faculty of Engineering at Aden University, Yemen. The author defines Cloud computing as a new emerging technology and trends around the world now. He said cloud computing applications are running as a service over the internet on scalable infrastructure. Universities can take advantage of cloud computing application to provide students and teachers with free or low cost tools. His work focused on the students awareness of cloud computing and how often they use cloud computing application. The work investigated the student's opinion of using cloud computing collaboration application and the university administrator support to integrate that new technology. The finding therefore recommends education and sensitization on cloud computing in order to increase awareness and knowledge about this emerging technology and its prospects [15].

In his thesis titled *Distributed Pair Programming in Global Software Development,* [16]observed that Global software development has become a widespread business practice over the course of recent years. Organizations are seeking to apply agile methodologies to the globally distributed environment. However, the principle of close collaboration as applied in agile methodologies remains in conflict with the present trends favoring the development of software in geographically distributed teams. Distributed pair programming (DPP) is one of the most important and commonplace practices in distributed agile software development. It supports two developers working on the same task in different locations. To resolve the collaboration and other issues related to DPP, researchers have proposed varied approaches to increase the flexibility of pair programming in the distributed environment to make it more widely available to the increasingly large numbers of disparate users. In his work, the analyses and evaluation of the current situation as related to DPP was done first, including analyzing social practices associated with DPP and evaluating several existing DPP tools. The final analysis of his results shows that learning or knowledge sharing has been recognized as the biggest issue in regards to DPP and this is a new research area in the area of DPP.

According to the unique features of the learning process associated with DPP, two high level design principles have been created for the purpose of designing new learning functions, to be applied to the existing tools: Active learning design principle and Pair formation pattern design principle. The Active Learning design principle is intended to improve the learning from active learning theory and the pair formation pattern design principle is intending to improve the learning from practical pair formation pattern in relation to DPP. XPairtise, a widely DPP tool, was chosen as the basis of a framework upon which the new learning functions was based. Finally, the results of the experiment show that pairs with different backgrounds can achieve better learning results by using the new tools, than if they were using the traditional tools.

[17] reported in their work titled "Enabling Distributed Pair Programming in Eclipse" that Distributed pair programming is still a challenging field for tool developers. In their work, they analyze the social interaction in distributed pair programming and investigate how current technology supports this interaction. In addition, XPairtise was also presented, which is a plug-in for eclipse that allows instant pair programming between distributed development teams. In their future work, they plan to include ACTIVITY INDICATORS or EXPERT FINDER mechanisms as they were, e.g., present in TUKAN that allow to retrieve experts from the registered users for specific problem domains. This retrieval could, e.g., be based on source code analysis or activity analysis of pair programming session. For improving the teaching functionality, they planned to include a REPLAY mechanism which records pair programming session and allows to review them afterwards. As at then, they are about organising controlled field experiment in order to evaluate the use of XPairtise in further lab courses and better understand the impact of the plugin on distributed team performance.

In a work titled "Evaluating Tools that Support Pair Programming in a Distributed Engineering Environment", [17] opines that the construction and improvement of high-quality products in a global software development environment requires flexible practices for collaboration and a tool that support these practices in a distributed software development team. Pair Programming (PP), a well-known agile practice has been reported to improve software productivity and software quality in co-located environments. They however said that PP needs stronger tool support to address new challenges such as communication, distributed collaboration, and data exchange in distributed environment. Their work introduces a systematic tool evaluation approach for distributed pair programming (DPP) and a reports on an initial tool survey of the available open source tools. Their major findings were that DPP was not fully supported by any tool under investigation and that some tools are limited to selective and individual aspects of DPP requirements. They concluded that the results of the tool evaluation can help project managers in selecting adequate tools and tool developers in providing high-value features for better support of DPP as basis for improving the quality of distributed engineering projects [18].

A work titled "*A Development Environment for Distributed Synchronous Collaborative Programming*" by [19] said that collaborative approaches in the classroom have been shown to be highly beneficial for students of computer science; obstacles inherent in today's academic environment often prevent collocated collaborative approaches from being implemented. They are of the opinion that a solution to the collocation problem may lie with tools that facilitate distributed collaboration. Therefore, presented RIPPLE Remote Interactive Pair Programming and Learning Environment), a development environment for

distributed synchronous collaborative programming. RIPPLE is an open source software tool. Their initial user tests demonstrate positive responses from students, and also show that potential for long term learning, motivation, and retention benefits is significant. In addition to its benefits for students, RIPPLE is a tool for computing education researchers who wish to collect data on collaborative programming. RIPPLE is a plug-in for the popular Eclipse integrated development environment. When two users work together remotely through RIPPLE, each executes a separate local installation of Eclipse with the RIPPLE plug-in. From each programmer's perspective, there are two panes: the program development pane and the textual dialogue pane.

3.0 DATA ANALYSIS STRATEGY

This research employed Qualitative data analysis techniques in its data analysis. R statistical tool was used in the analysis. R was chosen because of its robustness and openness. It has the capacity to handle a variety of specific statistical analyses as well as interactive programming language. Analysis of Variance (ANOVA) which is a rigorous statistical tool used in making inferential decisions in experimental design studies to ensure the equivalence of comparative groups even when number per group differed across the group. Some of the data obtained in this study were also subjected to independent samples t-test at $p = 0.05$. Descriptive statistics were also employed in the analysis of the data.

4.0 DISCUSSION OF RESULT

Prior to conducting a preliminary experiment, a pre-experimental survey carried out to know level of knowledge and involvement of the subjects in pair programming and distributed pair programming. The result is as presented in Table 5.1-5.8.

Summarily, Table (5.3) above shows the percentage of Programming Languages (PL) that respondents are familiar with. Java programming language is the most populous among respondent and has the highest percentage of 74.51%. Table (5.4) shows that 68.6% of the respondents have heard of Pair programming approach to software development which make them suitable for the experiment.

Table 5.1: Age bracket of the respondent

Age	Number	Percentage
15-20	19	37.3
21-30	28	54.9
Above 30	4	7.8
Total	**51**	**100.0**

Table 5.2: Status of the respondent

Professional Status	Number
Student	49
Software manager	1
Software engineer	1
Total	**51**

Table 5.3: Programming languages (PL) used by the respondent

PL familiar with	Number	Percentage
Java	38	74.51
PHP	5	9.80
ASP.NET	3	5.88
Others	5	9.80
Total	**51**	**99.99**

Table 5.4: Number of the respondents who have heard of PP

Heard of PP	Number	Percentage
Yes	35	68.6
No	16	31.4
Total	**51**	**100.0**

Table 5.5: Number of the respondents who have participated in PP

Participation in PP	Number	Percentage
Yes	18	35.3
No	33	34.7
Total	51	**70.00**

Table 5.6: Number of the respondents who have heard of distributed software development

Aware of Distributed Software Development	Number	Percentage
Yes	25	51.0
No	26	49.0
Total	**51**	**100.0**

Table 5.7: Number of the respondents who have participated in distributed software development

Participated in DSD	Number	Percentage
Yes	5	9.8
No	46	90.2
Total	**51**	**100.0**

Table 5.8: Tool used by the respondent in distributed software development

Type of Tool	Number	Percentage
IDE	5	9.8
None	46	90.2
Total	**51**	**100.0**

Table (5.5) above shows that 35.3 % of the respondents have participated in pair programming before this experiment while only 9.8% (from Table5.7) of the respondents have heard of Distributed pair programming before the experiment. Table (5.8) above shows that only 5% of respondents have participated in Distributed software development using Integrated Development Environment (IDE).

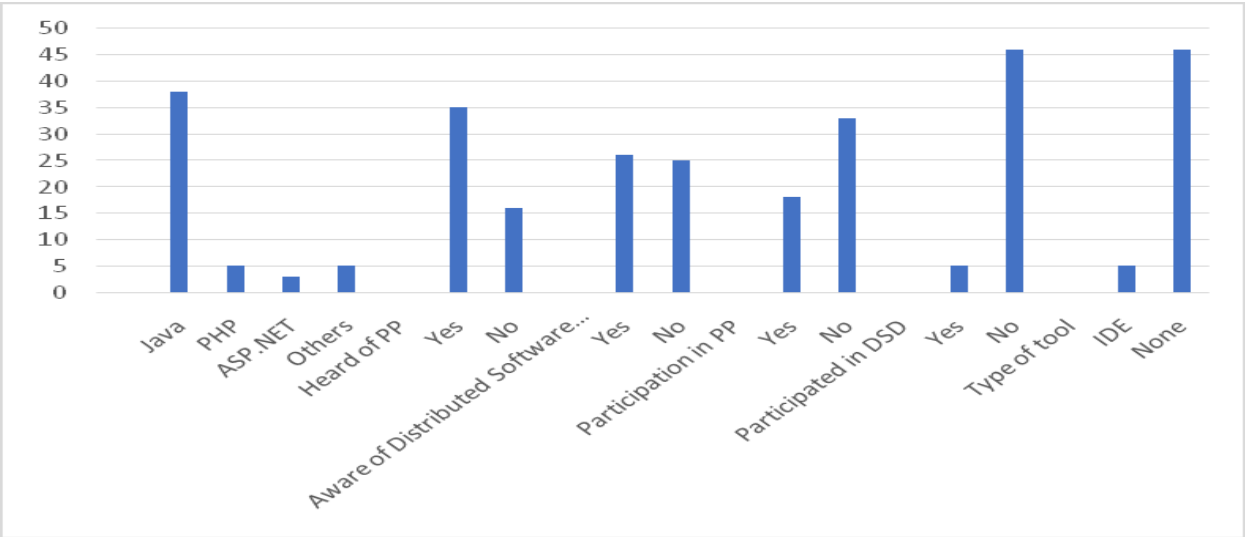

Fig 1: The summary of the survey result

Figure 1 shows the summary of the result gotten from the pilot survey. It shows that majority of the students that took part in the experiment were familiar with java programming language. Thirty-five (35), which amount to 68.6% has heard of pair programming before while 18 (35.3%) have actually participated in pair programming. Also, 51% of the participants are aware of distributed pair programming where only 5% have participated in distributed pair programming before using an IDE.

5.0 CONCLUSION

The purpose of this work is to evaluate the student's awareness and involvement regarding distributed computing and Agile methodology. For this purpose a survey was conducted through online questionnaires to evaluate the objectives from students of computer science, university of Ibadan, Nigeria. Findings show that most of the students were not familiar with practices due to lack of resources though some of them have heard of it.Despite the increase in software process improvement research especially the area of distributed software development, this study concluded that the level of awareness of the practices is very low among the student of computer science. Also the use of the DPP tools among the computer sciences student is low. There is therefore the need to focus and raise awareness on its benefits and importance in teaching system to extend the lack in computing resources. Introduction of distributed software development and agile methodology in the University curriculum is highly necessary so that students can take it as a course.

5.1 Future Work

In the future, we intend to develop a tool that can be applicable Nigeria environment for the research and teachings distributed pair programming. The tool shall be used in a controlled experiment over a period of time to determine its effectiveness.

ACKNOWLEDGEMENT

The researcher appreciates the voluntary and dedication of the students that participated in this study.

REFERENCES

[1] N. Mohanraj and. Sankar, A Distributed Pair Programming: A Survey. *International Journal of Engineering Research and Technology (IJERT.)* ISSN:2278-0181 Vol. 3 Issue 8, 2014 .

[2] S. O. Akinola, An Empirical Comparative Analysis of Programming Effort, Bugs Incurrence and Code Quality between Solo and Pair Programmers, *Middle-East Journal of Scientific Research,* 21 (12): 2231-2237, 2014 ISSN 1990-9233 © IDOSI Publications, 2014 DOI: 10.5829/idosi.mejsr.2014.21.12.21843

[3] W. Dietmar, B. Stefan, K. Andreas, Evaluating tools that support pair programming in a distributed engineering environment. *In Proc. Conference on Evaluation and Assessment in Software Engineering (EASE)*, pages 1–10. Vortrag: Proc. Conference on Evaluation and Assessment in Software Engineering (EASE), Keele, Great Britain; 2010-04-12–2010-04-13.

[4] M.M. Müller, Do Programmer Pairs make different Mistakes than Solo Programmers*? Proc of 10th Int. Conf on Evaluation and Assessment in Software Engineering EASE)*, 2006, Keele University.

[5] F. Padberg, M. M Müller. *Analyzing the cost and benefit of pair programming*, Proc of 9th Metrics Symp, pp.166-177. 2003.

[6] Williams L., Kessler R. R., Cunningham W. and Jeffries R., 2000. Strengthening the Case for Pair Programming. IEEE Software 17(4) pages 19–25.

[7] K. Schwaber, M. Beedle Agile Soft. Develop-ment with Scrum. Prentice-Hall, 2002.

[8] M. Paasivaara, S. Durasiewicz, C. Lassenius, "Using Scrum in a globally distributed project: A case study," Soft. Process Improvement and Practice, vol. 13, no. 6, pp. 527–544, 2008.

[9] G. Hanssen, D. Smite, N. Moe, "Signs of Agile trends in Global Soft. Eng. research: A tertiary study," in Intl. Conf. on Global Soft. Eng. Workshop ICGSEW), pp. 17 –23, 2011.

[10] E. Hossain, M. A. Babar, H.-y. Paik, "Using scrum in global soft. development: A systematic literature review," in Proc. of the 4th IEEE Intl. Conf. on Global Soft. Eng., pp. 175–184, 2009.

[11] D. Damian, A. Hadwin, and B. Al-Ani, "Instructional design and assessment strategies for teaching global software development: a framework," in Proc. of the 28th Intl. Conf. on Soft. Eng., pp. 685–690, (2006).

[12] W. Aspray, F. Mayadas, M. Y. Vardi, "Globalization and offshoring of software," Report of the ACM Job Migration Task Force, Association for Computing Machinery, New York, NY, USA, 2006.

[13] Miguel Jimenez, Mario Piattini, Aurora Vizcaıno, Challenges and Improvements in Distributed Software Development: A System-atic Review, Hindawi Publishing Corporation *Advances in Software Engineering*, Volume 2009, Article ID 710971, 14 pages doi:10.1155/2009/ 710971

[14] R. Davison, "Offshoring information techno-logy: sourcing and outsourcing to a global workforce," *Information Technology for Deve-lopment*, vol. 13, no. 1, pp. 101–102, 2007.

[15] S. B. A. Samah, Student's Awareness of Cloud Computing : Case Study Faculty of Engineering at Aden University , Yemen, (*2*1), 1122–1129, (2014).

[16] Jingchun Hao, (2011). Distributed Pair Programming in Global Software Development, A dissertation submitted to School of Informatics, University of Edinburgh. Available on http://www.inf.ed.ac.uk/publications/thesis/online/IM111131.pdf.

[17] Schuemmer T. and Stephan L., October 2009. Understanding tools and practices for distributed pair programming. *Journal of Universal Computer Science, vol. 15, no. 16 pages* 3101–3125.

[18] D. Winkler, S. Biffl, A. Kaltenbach, Evaluating Tools that Support Pair Programming in a Distributed Engineering Environment. *International Conference on Evaluation and Assessment in Software Engineering EASE (2010)*, 1–10.

[19] K. E. Boyer, A. Dwight, R. T. Fondren, M. A Vouk, J.C. Lester. A development environment for distributed synchronous collaborative programming. *ACM SIGCSE Bulletin*, *40(*3), 158. https://doi.org/10.1145/1597849.1384315, (2008).

[20] M. Monasor, A. Vizcaıno, M. Piattini, I. Caballero, "Preparing students and engineers for global soft. Development: A systematic review," in Intl. Conf. on Global Soft. Eng., pp. 177 –186. 2010.

[21] A. Sivakumar, G. Singaravelu, Utilization of Cloud Computing Application. *American Journal of Educational Research*, *4*1(1), 792-797, 2016.